

Banco de Dados



Dependências Funcionais e Normalização de Bancos de Dados Relacionais

João Eduardo Ferreira
Osvaldo Kotaro Takai

Abordagens de Projeto

□ Top-down

- Iniciar com o agrupamento dos atributos obtidos a partir do projeto conceitual de mapeamento
- Isso é chamado de projeto por análise

□ Bottom-up

- Considerar os relacionamentos entre atributos
- Construir as relações
- Isso é chamado projeto pela síntese

□ Nossa Abordagem:

- Utilizar a abordagem Top-down para obter as relações
- Utilizar a abordagem Bottom-up para melhorar a qualidade das relações obtidas anteriormente

Objetivos

1. Apresentar um Guia Informal para projetos de Bancos de Dados Relacionais
 - Semântica de Atributos
 - Informação redundante em tuplas e Anomalias de Atualização
 - Valores Nulos em tuplas
 - Tuplas Espúrias

Objetivos

1. Apresentar um guia formal para avaliar a qualidade das relações de um banco de dados
 - Dependências Funcionais (DFs)
 - Definição de DF
 - Regras de inferências para DFs
 - Formas Normais
 - Normalização de Relações
 - Uso prático das Formas Normais
 - Definição de Chaves e Atributos que participam das Chaves
 - Primeira Forma Normal
 - Segunda Forma Normal
 - Terceira Forma Normal
 - Definição Geral de Forma Normal (Múltiplas Chaves)
 - BCNF (Boyce-Codd Normal Form)

Guia Informal

- ❑ Projetar um banco de dados relacional significa:
 - Agrupar atributos para formar “bons” esquemas de relações
 - O que é um bom esquema?
 - ❑ No nível lógico
 - É fácil de entender
 - Ajuda a formular consultas corretas
 - ❑ No nível físico
 - Tuplas são armazenadas eficientemente
 - Tuplas são acessadas com eficiência
- ❑ Existem 2 níveis de esquemas de relações
 - Nível lógico (visão do usuário)
 - Nível de armazenamento (relações base)
- ❑ O projeto considera as relações base

Guia Informal: Semântica de Atributos

- ❑ **Guia 1:** Informalmente, cada tupla de uma relação deve representar uma entidade ou instância de relacionamento. Assim:
 - Atributos de entidades distintas não devem estar na mesma relação
 - Apenas chaves-estrangeiras devem ser usadas para referenciar outras entidades
 - Atributos de entidades e relacionamentos devem ser mantidos separadamente tanto quanto possível
 - **Dica:** Projete um esquema que possa ser facilmente explicada. A semântica dos atributos deve ser de fácil interpretação

Guia Informal: Semântica de Atributos

□ Exemplo

■ O que é mais claro?

- Empregado(enome, nss, datanasc, endereço, dnumero)
- Projeto(pnome, pnumero, plocalizacao, dnum)
- Trabalha-em(nss, púmero, horas)

Ou

- EMP_PROJ (nss, púmero, horas, enome, pnome, plocalizacao)

Guia Informal: Semântica de Atributos

□ Exemplo

■ Qual esquema necessita de maior espaço de armazenamento?

- Empregado(enome, nss, datanasc, endereço, dnumero)
- Projeto(pnome, pnumero, plocalizacao, dnum)
- Trabalha-em(nss, pnúmero, horas)

Ou

- EMP_PROJ (nss, pnúmero, horas, enome, pnome, plocalizacao)

Guia Informal: Informação redundante em tuplas e Anomalias de Atualização

- ❑ Informações redundantes desperdiçam o espaço de armazenamento
- ❑ A mistura atributos de várias entidades pode gerar problemas conhecidos como anomalias de atualização
 - Anomalias de inserção
 - Anomalias de remoção
 - Anomalias de modificação

Exemplo de Anomalia de Atualização

- ❑ Considere a relação:
 - EMP_PROJ (nss, pnúmero, horas, enome, pnome, plocalizacao)
- ❑ Anomalia de Atualização:
 - A mudança de nome do projeto de número 20 de 'Faturamento' para 'Conta-Cliente' provoca alterações em todos os empregados que trabalham nesse projeto

Exemplo de Anomalia de Atualização

❑ Anomalia de Inserção:

- Não se pode inserir um projeto a menos que um empregado esteja associado. Por outro lado, não se pode inserir um empregado a menos que ele esteja associado a um projeto

❑ Anomalia de Remoção:

- Quando um projeto é removido, todos os empregados que trabalham no projeto é removido. Alternativamente, se um empregado for o único empregado do projeto, a remoção desse empregado resultará na remoção do projeto correspondente

Informação redundante em tuplas e Anomalias de Atualização

- ❑ **Guia 2:** Projete um esquema que não sofra de anomalias de inserção, remoção e de atualização. Se existir alguma, então assegure-se de que as aplicações levem tais anomalias em consideração

Valores Nulos em Tuplas

- ❑ **Guia 3:** Relações devem ser projetadas de forma que suas tuplas tenham a menor quantidade possível de valores nulos
 - Normalmente os atributos que possuem valores nulos podem ser colocados em relações separadas (com uma chave-primária)
- ❑ Razões para os valores nulos:
 - Valor não aplicável ou inválido
 - Valor desconhecido (embora possa existir)
 - Valor indisponível (embora se saiba que exista)

Tuplas Espúrias

- ❑ Projetos incorretos de BDRs podem gerar resultados inválidos em certas operações Join
- ❑ A propriedade de “junção sem perdas” é usada para garantir resultados corretos em operações Join
- ❑ **Guia 4:** As relações devem ser projetadas para satisfazer a condição de junção sem perdas. Nenhuma tupla espúria deve ser gerada ao fazer um join natural de qualquer relação

Tuplas Espúrias

- ❑ Existem duas propriedades importantes de decomposições:
 - Não-aditiva ou sem perdas (losslessness) do join correspondente
 - Preservação das dependências funcionais
- ❑ Note que a propriedade (a) é extremamente importante e não pode ser sacrificada. A propriedade (b) é menos restrito e pode ser sacrificada

O exemplo de tuplas espúrias na figura 10.6, página 217 do livro Elmasri & Navathe apresenta um erro. Não existe indicação com asteriscos das tuplas espúrias!

	<u>NSS</u>	<u>PNÚMERO</u>	<u>HORAS</u>	<u>PNOME</u>	<u>PLOCALIZAÇÃO</u>	<u>ENAME</u>
	123456789	1	32.5	ProdutoX	Bellaire	John Smith
*	123456789	1	32.5	ProdutoX	Bellaire	Joyce English
	123456789	2	7.5	ProdutoY	Sugarland	John Smith
*	123456789	2	7.5	ProdutoY	Sugarland	Joyce English
*	123456789	2	7.5	ProdutoY	Sugarland	Franklin Wong
	666884444	3	40.0	ProdutoZ	Houston	Ramesh Narayan
*	666884444	3	40.0	ProdutoZ	Houston	Franklin Wong
*	453453453	1	20.0	ProdutoX	Bellaire	John Smith
	453453453	1	20.0	ProdutoX	Bellaire	Joyce English
*	453453453	2	20.0	ProdutoY	Sugarland	John Smith
	453453453	2	20.0	ProdutoY	Sugarland	Joyce English
*	453453453	2	20.0	ProdutoY	Sugarland	Franklin Wong
*	333445555	2	10.0	ProdutoY	Sugarland	John Smith
*	333445555	2	10.0	ProdutoY	Sugarland	Joyce English
	333445555	2	10.0	ProdutoY	Sugarland	Franklin Wong
*	333445555	3	10.0	ProdutoZ	Houston	Ramesh Narayan
	333445555	3	10.0	ProdutoZ	Houston	Franklin Wong
	333445555	10	10.0	Automação	Stafford	Franklin Wong
*	333445555	20	10.0	Reorganização	Houston	Ramesh Narayan
	333445555	20	10.0	Reorganização	Houston	Franklin Wong

Dependências Funcionais

- ❑ Dependências funcionais (DFs) são usadas para medir formalmente a qualidade do projeto relacional
- ❑ As DFs e chaves são usadas para definir formas normais de relações
- ❑ As DFs são restrições que são derivadas do significado e do inter-relacionamento dos dados de atributos
- ❑ Um conjunto de atributos X determina funcionalmente um conjunto de atributos Y se o valor de X determinar um único valor Y

Dependências Funcionais

- $X \rightarrow Y$ diz que se duas tuplas tiverem o mesmo valor para X , elas devem ter o mesmo valor para Y . Ou seja:
 - Se $X \rightarrow Y$ então, para quaisquer tuplas t_1 e t_2 de $r(R)$:
Se $t_1[X] = t_2[X]$, então $t_1[Y] = t_2[Y]$
- Se K é uma chave de R , então K determina funcionalmente todos os atributos de R (uma vez que nós nunca teremos duas tuplas distintas com $t_1[K] = t_2[K]$)
- **Importante:**
 - $X \rightarrow Y$ especifica uma restrição sobre todas as instâncias de $r(R)$
 - As DFs são derivadas das restrições do mundo real e não de uma extensão específica da relação R

Exemplos de Restrições de DF

- ❑ O número do seguro social determina o nome do empregado
 - $NSS \rightarrow ENOME$
- ❑ O número do projeto determina o nome do projeto e a sua localização
 - $PNUMERO \rightarrow \{ PNAME, PLOCALIZACAO \}$
- ❑ O nss de empregado e o número do projeto determinam as horas semanais que o empregado trabalha no projeto
 - $\{ NSS, PNUMERO \} \rightarrow HORAS$

Regras de Inferência para DFs

- ❑ Regras de inferência de Armstrong:
 - RI1. (Reflexiva) Se Y é subconjunto de X , então $X \rightarrow Y$
(Isso também é válido quando $X=Y$)
 - RI2. (Aumentativa) Se $X \rightarrow Y$, então $XZ \rightarrow YZ$
(Notação: XZ significa $X \cup Z$)
 - RI3. (Transitiva) Se $X \rightarrow Y$ e $Y \rightarrow Z$, então $X \rightarrow Z$
- ❑ RI1, RI2 e RI3 formam um conjunto completo de regras de inferência

Regras de Inferência para DFs

- Algumas regras de inferência úteis:
 - (Decomposição) Se $X \rightarrow YZ$, então $X \rightarrow Y$ e $X \rightarrow Z$
 - (Aditiva) Se $X \rightarrow Y$ e $X \rightarrow Z$, então $X \rightarrow YZ$
 - (Pseudotransitiva) Se $X \rightarrow Y$ e $WY \rightarrow Z$, então $WX \rightarrow Z$

- As três regras de inferência acima, bem como quaisquer outras regras de inferência, podem ser deduzidas a partir de RI1, RI2 e RI3 (propriedade de ser completa)

Formas Normais com base em Chaves Primárias

- ❑ Normalização de Relações
- ❑ Uso prático de Formas Normais
- ❑ Definições de Chaves e de Atributos que participam de Chaves
- ❑ Primeira Forma Normal
- ❑ Segunda Forma Normal
- ❑ Terceira Forma Normal

Normalização de Relações

- ❑ Normalização: Processo de decompor relações “ruins” dividindo seus atributos em relações menores
- ❑ Forma Normal: Indica o nível de qualidade de uma relação

Normalização de Relações (2)

- ❑ 2FN, 3FN, BCNF baseiam-se em chaves e DFs de uma relação esquema
- ❑ 4FN e 5FN baseiam-se em chaves e dependências multivaloradas (não serão discutidas)

Uso Prático das Formas Normais

- ❑ Na prática, a normalização é realizada para obter projetos de alta qualidade e atender às propriedades desejáveis
- ❑ Os projetistas de bancos de dados não precisam normalizar na maior forma normal possível.
- ❑ Desnormalização: processo de armazenar junções de relações de forma normal superior como uma relação base que está numa forma normal inferior

Definição de Chaves e Atributos que Participam de Chaves

■ Revisão:

- Uma superchave de uma relação esquema $R = \{A_1, A_2, \dots, A_n\}$ é um conjunto de atributos S , subconjunto de R com a propriedade de que $t_1[S] \neq t_2[S]$ para qualquer extensão $r(R)$
- Uma superchave K é uma chave se K é uma superchave mínima
- Se uma relação esquema tiver mais de uma chave, cada chave será chamada de chave-candidata. Uma das chaves-candidatas é arbitrariamente escolhida para ser a chave-primária e as outras são chamadas de chaves-secundárias

Definição de Chaves e Atributos que Participam de Chaves

- ❑ Um **atributo primo** (ou primário) é membro de alguma chave-candidata
- ❑ Um **atributo não-primo** é um atributo que não é primo – isto é, não é membro de qualquer chave-candidata

Primeira Forma Normal

- ❑ Proíbe atributos compostos, atributos multivalorados e relações aninhadas. Ou seja, permite apenas atributos que sejam atômicos
- ❑ Considerado como sendo parte da definição de relação

Normalização na 1 FN

Normalization into 1NF. (a) Relation schema that is not in 1NF. (b) Example relation instance. (c) 1NF relation with redundancy.

(a)

DEPARTMENT			
DNAME	<u>DNUMBER</u>	DMGRSSN	DLOCATIONS

(b)

DEPARTMENT			
DNAME	<u>DNUMBER</u>	DMGRSSN	DLOCATIONS
Research	5	333445555	{Bellaire, Sugarland, Houston}
Administration	4	987654321	{Stafford}
Headquarters	1	888665555	{Houston}

(c)

DEPARTMENT			
DNAME	<u>DNUMBER</u>	DMGRSSN	<u>DLOCATION</u>
Research	5	333445555	Bellaire
Research	5	333445555	Sugarland
Research	5	333445555	Houston
Administration	4	987654321	Stafford
Headquarters	1	888665555	Houston

© Addison Wesley Longman, Inc. 2000, Elmasri/Navathe, Fundamentals of Database Systems, Third Edition

Ver figura 10.8, página 225 do livro Elmasri & Navathe

Normalização de relações Aninhadas para a 1 FN

Normalizing nested relations into 1NF. (a) Schema of the EMP_PROJ relation with a “nested relation” PROJS. (b) Example extension of the EMP_PROJ relation showing nested relations within each tuple. (c) Decomposing EMP_PROJ into 1NF relations EMP_PROJ1 and EMP_PROJ2 by propagating the primary key.

(a)

EMP_PROJ		PROJS	
SSN	ENAME	PNUMBER	HOURS

(b)

SSN	ENAME	PNUMBER	HOURS
123456789	Smith, John B.	1	32.5
		2	7.5
666884444	Narayan, Ramesh K.	3	40.0
453453453	English, Joyce A.	1	20.0
		2	20.0
333445555	Wong, Franklin T.	2	10.0
		3	10.0
		10	10.0
		20	10.0
999887777	Zelaya, Alida J.	30	30.0
		10	10.0
987987987	Jabbar, Ahmad V.	10	35.0
		30	5.0
987654321	Wallace, Jennifer S.	30	20.0
		20	15.0
888665555	Borg, James E.	20	null

(c)

EMP_PROJ1	
SSN	ENAME

EMP_PROJ2	
SSN	PNUMBER

EMP_PROJ2	
PNUMBER	HOURS

© Addison Wesley Longman, Inc. 2000, Elmasri/Navathe, Fundamentals of Database Systems, Third Edition

Ver figura 10.9, página 226 do livro Elmasri & Navathe

Segunda Forma Normal

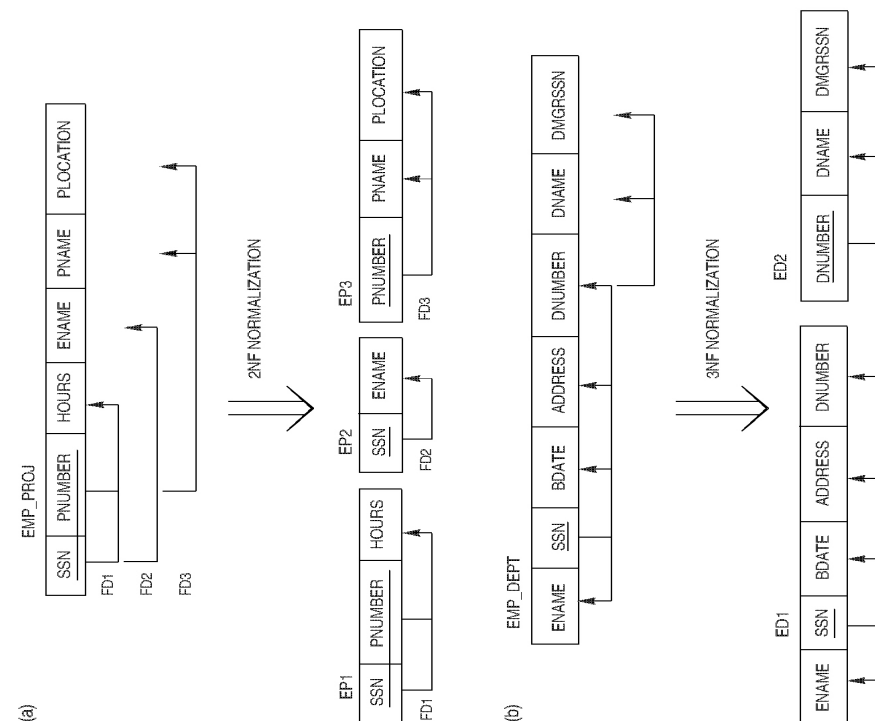
- ❑ Utiliza conceitos de DFs e chave-primária
 - ❑ Definições utilizadas:
 - **Atributo Primo** – atributo que é membro da chave primária K
 - **Dependência funcional total** – uma DF, $Y \rightarrow Z$, onde a remoção de qualquer atributo de Y invalida a DF.
- Exemplos:
- ❑ $\{ \text{NSS}, \text{PNUMERO} \} \rightarrow \text{HORAS}$ é uma DF total, uma vez que NSS não determina HORAS e nem PNUMERO determina HORAS
 - ❑ $\{ \text{NSS}, \text{PNUMERO} \} \rightarrow \text{ENOME}$ não é uma DF total (é uma DF parcial) pois $\text{NSS} \rightarrow \text{ENOME}$

Segunda Forma Normal

- ❑ Uma relação esquema R está na 2FN se estiver na 1FN e todos os atributos não-primos A de R forem totalmente dependentes da chave-primária
- ❑ R pode ser decomposto em relações que estejam na 2 FN através do processo de normalização

Normalização para a 2FN e 3FN

The normalization process. (a) Normalizing EMP_PROJ into 2NF relations. (b) Normalizing EMP_DEPT into 3NF relations.



© Addison Wesley Longman, Inc. 2000, Elmasri/Navathe, Fundamentals of Database Systems, Third Edition

Ver figura 10.10, página 228 do livro Elmasri & Navathe

Terceira Forma Normal

□ Definição:

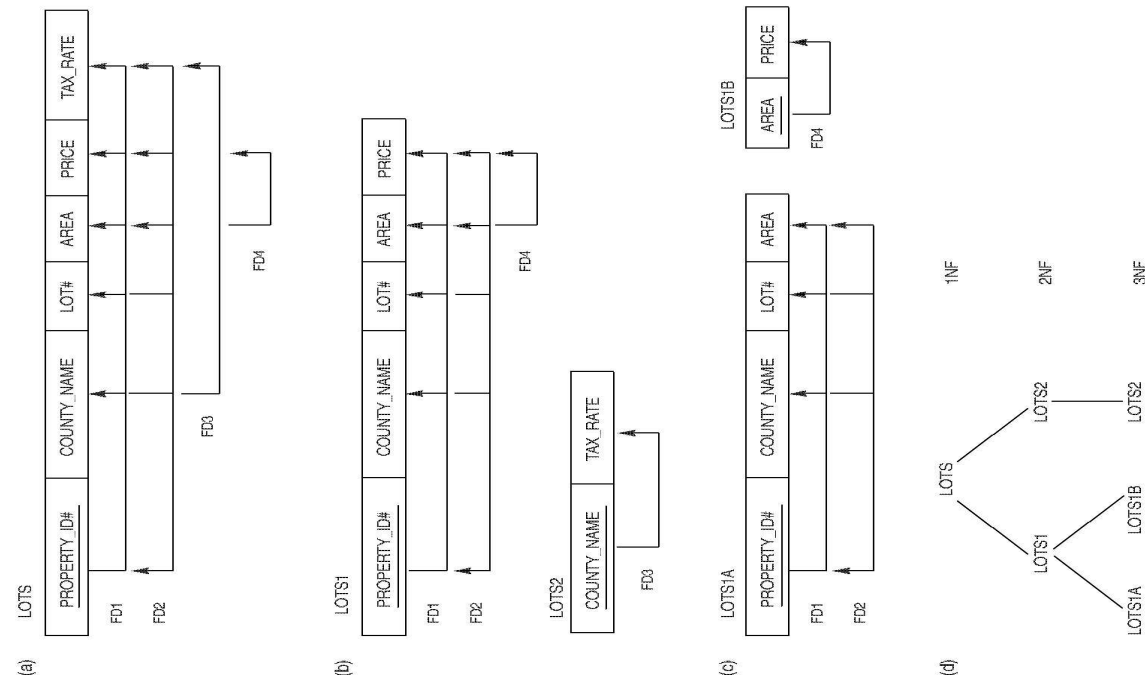
- Dependência funcional transitiva – uma DF $X \rightarrow Z$ pode ser derivada a partir de duas DFs $X \rightarrow Y$ e $Y \rightarrow Z$.
- Exemplos:
 - Na figura anterior, $NSS \rightarrow NSSGER$ é uma DF transitiva pois $NSS \rightarrow DNUMERO$ e $DNUMERO \rightarrow NSSGER$
 - $NSS \rightarrow ENOME$ não é transitiva pois não existe um conjunto de atributos X onde $NSSN \rightarrow X$ e $X \rightarrow ENOME$

Terceira Forma Normal

- ❑ Uma relação esquema R está na 3FN se ela estiver na 2FN e nenhum atributo não-primário, A, for transitivamente dependente da chave-primária
- ❑ R pode ser decomposto em relações que estejam na 3FN via o processo de normalização
- ❑ NOTA:
 - Em $X \rightarrow Y$ e $Y \rightarrow Z$, sendo X a chave-primária, pode ser considerado um problema se, e somente se, Y não for uma chave-candidata. Quando Y é uma chave-candidata, não existe problema com a dependência transitiva
 - Por exemplo, considere EMP (NSS, Emp#, Salario).
 - ❑ Aqui, $NSS \rightarrow Emp\# \rightarrow Salario$ e Emp# é uma chave-candidata

Normalização para a 2FN e 3FN

Normalization to 2NF and 3NF. (a) The lots relation schema and its functional dependencies fd1 through fd4. (b) Decomposing lots into the 2NF relations LOTS1 and LOTS2. (c) Decomposing LOTS1 into the 3NF relations LOTS1A and LOTS1B. (d) Summary of normalization of lots.



© Addison Wesley Longman, Inc. 2000, Elmasri/Navathe, Fundamentals of Database Systems, Third Edition

Ver figura 10.11, página 230 do livro Elmasri & Navathe

Definição Geral de Formas Normais

- ❑ As definições anteriores consideravam somente a chave-primária
- ❑ As próximas definições levarão em consideração as várias chaves candidatas

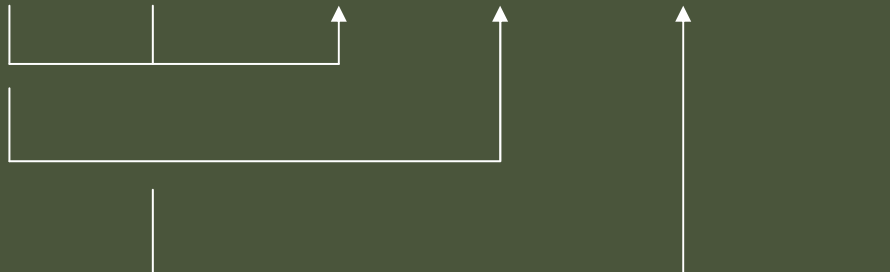
Definição Geral de Formas Normais

□ Redefinição da 2FN:

- Uma relação esquema R está na 2FN se todos os atributos não-primos, A, forem totalmente dependentes de todas as chaves de R

□ Teste:

- Verifique que EMP_PROJ não está na 2FN
 - EMP_PROJ (nss, pnúmero, horas, enome, pnome, plocalizacao)



Ver figura 10.3, página 213 do livro Elmasri & Navathe

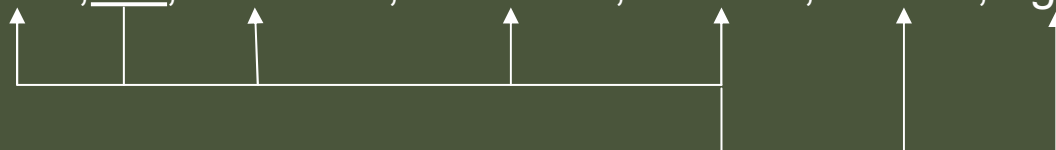
Definição Geral de Formas Normais

□ Redefinição de 3FN:

- Uma relação esquema R está na 3FN se, sempre que houver uma DF $X \rightarrow A$, então uma das duas condições são válidas:
 - X é uma superchave de R, ou
 - A é um atributo primo de R
- **NOTA:** A Forma normal de Boyce-Codd não admite a segunda condição

□ Teste:

- Verifique que está na 2FN mas não na 3FN
 - EMP_DEPT(enome,nss, datanasc, endereco, dumer, dnome, dgernss)



BCNF (Boyce-Codd Normal Form)

□ Definição de BCNF:

- Uma relação esquema R está na BCNF se, sempre que houver uma DF $X \rightarrow A$ em R, então X é uma superchave de R

□ Cada FN engloba a FN anterior:

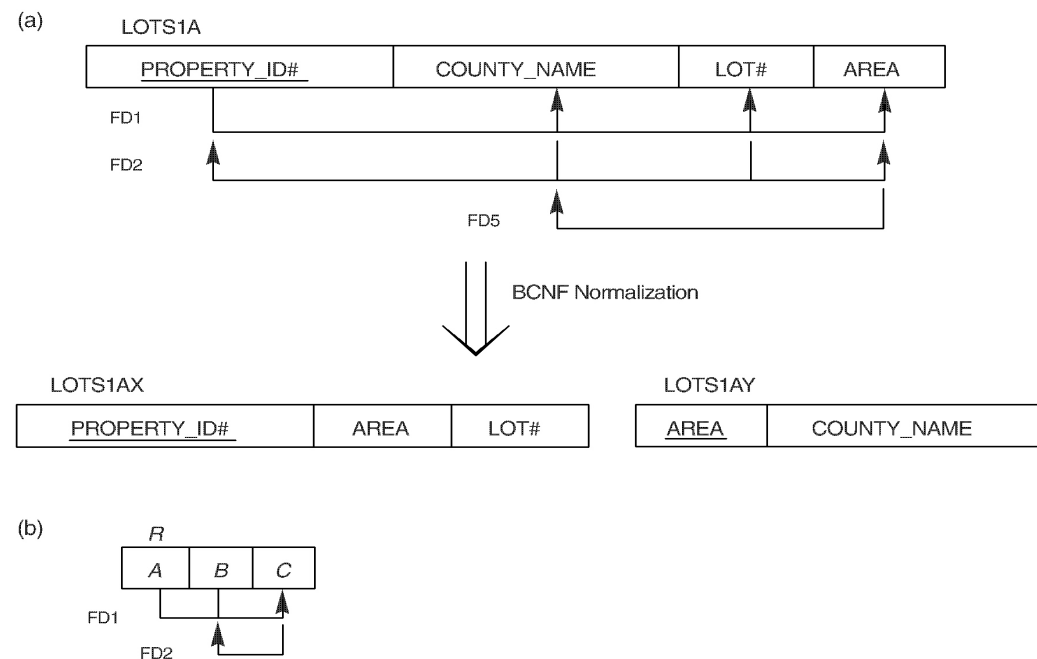
- Toda relação em 2FN está na 1FN
- Toda relação em 3FN está na 2FN
- Toda relação em BCNF está na 3FN

□ Existem relações que estão na 3FN mas não em BCNF

□ A meta é alcançar a BCNF (ou 3FN) em todas as relações

Boyce-Codd normal form

Boyce-Codd normal form. (a) BCNF normalization with the dependency of FD2 being “lost” in the decomposition.
(b) A relation R in 3NF but not in BCNF.



© Addison Wesley Longman, Inc. 2000, Elmasri/Navathe, Fundamentals of Database Systems, Third Edition

Ver figura 10.12, página 232 do livro Elmasri & Navathe

Relação em 3FN mas não em BCNF

A relation TEACH that is in 3NF but not in BCNF.

STUDENT	COURSE	INSTRUCTOR
Narayan	Database	Mark
Smith	Database	Navathe
Smith	Operating Systems	Ammar
Smith	Theory	Schulman
Wallace	Database	Mark
Wallace	Operating Systems	Ahamad
Wong	Database	Omiecinski
Zelaya	Database	Navathe

© Addison Wesley Longman, Inc. 2000, Elmasri/Navathe, Fundamentals of Database Systems, Third Edition

Alcançando a BCNF pela Decomposição

- ❑ Existem duas DF em relação TEACH:
 - $df1: \{ \text{student, course} \} \rightarrow \text{instructor}$
 - $df2: \text{instructor} \rightarrow \text{course}$
- ❑ $\{ \text{student, course} \}$ é uma chave candidata da relação e esta dependência segue o padrão apresentado na figura 10.12(b). Assim esta relação está na 3FN mas não na BCNF
- ❑ Uma relação que não esteja na BCNF deve ser decomposta para atender a esta propriedade, enquanto possivelmente abdica da preservação de todas as dependências funcionais nas relações decompostas

Alcançando a BCNF pela Decomposição

- ❑ As três possíveis decomposições para relação TEACH
 - { student, instructor } e { student, course }
 - { course, instructor } e { course, student }
 - { instructor, course } e { instructor, student }
- ❑ Todas as três decomposições perdem a df1. Temos que conviver com este sacrifício, mas nós não podemos sacrificar a propriedade não-aditiva após a decomposição
- ❑ Das três, apenas a terceira decomposição não gera tuplas espúrias após o join (e assim, mantém a propriedade não-aditiva)
- ❑ Um teste para determinar se uma decomposição binária (decomposição em duas relações) é não-aditiva (lossless) é discutida na seção 11.1.4, página 241.

Questões

- Dado o DER de uma locadora de vídeo (próximo slide), e o mapeamento realizado para o esquema do BD Relacional, verifique a qualidade das relações obtidas (qual forma normal atingida) e, se necessário, normalize todos os esquemas de relações para a 3FN ou, se possível, para a BCNF.

